Task Arithmetic Through The Lens Of One-Shot Federated Learning

Anonymous authors
Paper under double-blind review

Abstract

Task Arithmetic is a model merging technique that enables the combination of multiple models' capabilities into a single model through simple arithmetic in the weight space, without the need for additional fine-tuning or access to the original training data. However, the factors that determine the success of Task Arithmetic remain unclear. In this paper, we examine Task Arithmetic for multi-task learning by framing it as a one-shot Federated Learning problem. We demonstrate that Task Arithmetic is mathematically equivalent to the commonly used algorithm in Federated Learning, called Federated Averaging (FedAvg). By leveraging well-established theoretical results from FedAvg, we identify two key factors that impact the performance of Task Arithmetic: data heterogeneity and training heterogeneity. To mitigate these challenges, we adapt several algorithms from Federated Learning to improve the effectiveness of Task Arithmetic. Our experiments demonstrate that applying these algorithms can often significantly boost performance of the merged model compared to the original Task Arithmetic approach. This work bridges Task Arithmetic and Federated Learning, offering new theoretical perspectives on Task Arithmetic and improved practical methodologies for model merging.

1 Introduction

With the proliferation of fine-tuned models across diverse domains, efficiently combining these models to achieve excellence across multiple tasks has emerged as a critical research challenge. *Task Arithmetic* (Ilharco et al., 2023), a recent technique in model merging, offers a simple yet effective solution. Consider a set of T tasks, each defined by a training dataset and a loss function for fine-tuning. Let $\theta_{\text{pre}} \in \mathbb{R}^d$ be the parameters of a pre-trained model and $\theta_t \in \mathbb{R}^d$ be the fine-tuned model parameters on task t. A *task vector* τ_t is generated by subtracting the pre-trained model parameters from the fine-tuned model parameters: $\tau_t = \theta_{\text{pre}} - \theta_t$. The sum of these task vectors, $\sum_{t=1}^{T} \tau_t$, produces a direction that enhances performance of the pre-trained model across multiple tasks for which the fine-tuned models have been trained. In other words, updating the pre-trained model parameters by $\theta_{\text{pre}} + \sum_{t=1}^{T} \tau_t$ results in a multi-task model. A key advantage of this approach is that it only involves element-wise operations in the weight space, eliminating the need for additional fine-tuning.

Despite its strong empirical performance, Task Arithmetic lacks substantial theoretical understanding. Only a small number of works have investigated this empirical success theoretically (Ortiz-Jimenez et al., 2023; Porrello et al., 2024). In this paper, we take a step towards bridging the gap between theory and practice by framing Task Arithmetic as a form of one-shot Federated Learning.

Federated Learning (McMahan et al., 2017), a distributed machine learning paradigm, enables devices to collaboratively train one shared model without exchanging the raw data. Federated Learning's goal is to retain data privacy and reduce computational costs, as all raw data remains stored locally on edge devices. In a typical Federated Learning training process, a server coordinates the training process by iterating through the following steps (Kairouz et al., 2021). First, the server broadcasts the current global model parameters and a training program to all the devices. Then each device locally computes an update to the model by using its own data. Finally, the server aggregates all the local updates from devices and updates the current

global model by using the aggregated local updates. A commonly used algorithm for this training process is Federated Averaging (FedAvg) (McMahan et al., 2017). In one-shot Federated Learning, the server learns a global model in only a single round of communication between itself and all the devices (Guha et al., 2019).

In this work, we show that using one-shot FedAvg is equivalent to Task Arithmetic, thus offering a new perspective on Task Arithmetic through the lens of one-shot Federated Learning. Using the connection between Federated Learning and Task Arithmetic, we can leverage the extensive theoretical and algorithmic advancements in Federated Learning to better understand when Task Arithmetic is effective and how it can be improved. To the best of our knowledge, this is the first study to bridge Federated Learning and Task Arithmetic. Our main contributions are summarized as follows.

Bridge Task Arithmetic and Federated Learning: We establish the connection between Task Arithmetic and one-shot Federated Averaging, formalizing Task Arithmetic using notions from Federated Learning.

Analyze the Impact of Data and Training Heterogeneity in Task Arithmetic: Data heterogeneity slows convergence in FedAvg, while training heterogeneity causes objective inconsistencies. We show that similar challenges exist in Task Arithmetic and analyze their impact using insights from Federated Learning, offering a deeper understanding of its convergence behavior.

Identify and Adapt Federated Learning Algorithms for Task Arithmetic: We identify and recommend Federated Learning algorithms to address heterogeneity challenges and enhance Task Arithmetic for better model merging performance.

Experimentally Show That Federated Learning Algorithms Often Improve Task Arithmetic: Experiments confirm that adapting Federated Learning algorithms often improves the merged model's performance compared to Task Arithmetic.

2 Related Work

As our work bridges Federated Learning and Task Arithmetic, a prominent approach within the growing domain of model merging, this section reviews related work on both model merging and Federated Learning.

Model Merging Task Arithmetic is one of many recent works on model merging (Wortsman et al., 2022a; Frankle et al., 2020b; Wortsman et al., 2022b; Matena & Raffel, 2022; Ainsworth et al., 2022; Don-Yehiya et al., 2022). Though the term model merging is relatively new, firstly formalized by Matena & Raffel (2022), the concept has received significant investigation (Wortsman et al., 2022b;a; Izmailov et al., 2018; Yang et al., 2019; Tarvainen & Valpola, 2017; Gupta et al., 2020). For example, Wortsman et al. (2022b) averages the pre-trained model parameters and fine-tuned parameters to enhance the robustness of fine-tuned model against distribution shifts. Wortsman et al. (2022a) averages parameters of multiple fine-tuned models with different hyperparameter configurations can improve robustness and accuracy. Izmailov et al. (2018) averages several parameters along the same trajectory of SGD can lead to better generalization.

Task Arithmetic, introduced by Ilharco et al. (2023), refines model merging by introducing task vectors and a hyperparameter λ to control how much task vectors modify pre-trained model parameters. This method has inspired various follow-up work on using simple arithmetic operations for model merging (Yu et al., 2024; Zhang et al., 2023; Ortiz-Jimenez et al., 2023; Stoica et al., 2023; Yadav et al., 2024; Stoica et al., 2023; Yang et al., 2023) such as sparsifying task vectors (Yu et al., 2024), merging parameter-efficient modules (Zhang et al., 2023), fine-tuning in linearized model spaces (Ortiz-Jimenez et al., 2023) and resolving sign interference of task vectors (Yadav et al., 2024).

A concurrent work of Task Arithmetic is Jin et al. (2022), who propose the *RegMean*. When merging two linear regression models, the model merging problem can be formulated as an optimization problem which has a closed-form solution. RegMean uses this solution to merge every linear layer of fine-tuned models. Unlike Task Arithmetic, RegMean is limited to linear layers and requires access to the input features of all linear layers. While Jin et al. (2022) also note that model merging is an extreme case of Federated Learning with single communication round, they do not explore this further. Moreover, RegMean focuses on merging linear layers, distinguishing it from our framework which addresses more general model merging problems.

There is a substantial body of research dedicated to understanding the effectiveness of model merging (Draxler et al., 2018; Entezari et al., 2021; Frankle et al., 2020a; Garipov et al., 2018; Benton et al., 2021; Li et al., 2018; Foret et al., 2020; Simsek et al., 2021). Some studies focus on the theory of linear model connectivity (Draxler et al., 2018; Entezari et al., 2021; Frankle et al., 2020a; Garipov et al., 2018), while others emphasize the flatness of the loss landscape (Benton et al., 2021; Li et al., 2018; Foret et al., 2020; Simsek et al., 2021). However, there has been relatively little work addressing the effectiveness of Task Arithmetic except for Ortiz-Jimenez et al. (2023); Porrello et al. (2024). Our work aims to bridge this gap.

Federated Learning In Federated Learning, *FedAvg* (McMahan et al., 2017) is widely used to solve the following distributed optimization problem across M devices: $\min_{\theta \in \mathbb{R}^d} L(\theta) := \frac{1}{M}\sum_{m=1}^{M} L_m(\theta)$ where $L_m(\theta) := \mathbb{E}_{x_m \sim \mathcal{D}_m}[\ell(\theta, x_m)]$ is the objective function on each device m, defined by some loss function ℓ and data distribution $\mathcal{D}_m$. The core idea behind *FedAvg* is to perform local Stochastic Gradient Descent (local SGD) on each device, followed by model averaging on the server. A substantial body of research has analyzed the performance of FedAvg and local SGD (Li et al., 2019; Karimireddy et al., 2020; Khaled et al., 2020; Koloskova et al., 2020; Woodworth et al., 2020b;a; Glasgow et al., 2022; Patel et al., 2024; Wang et al., 2022; Woodworth et al., 2021).

A key challenge in the theoretical analysis of FedAvg arises from data heterogeneity, where each device m has a different data distribution $\mathcal{D}_m$. In the homogeneous setting where $\mathcal{D}_m = \mathcal{D}\ \forall m$, Woodworth et al. (2021) have established the min-max complexity of FedAvg with smooth and convex loss functions. In the more complex heterogeneous setting, various works have derived the convergence rate of FedAvg under different assumptions about data heterogeneity (Li et al., 2019; Khaled et al., 2020; Koloskova et al., 2020; Woodworth et al., 2020b;a; Glasgow et al., 2022; Patel et al., 2024; Wang et al., 2022). In this work, rather than focusing on extending existing theoretical results, we focus on using these results to analyze Task Arithmetic.

To address the challenges posed by data heterogeneity, extensive research has focused on designing algorithms to improve the performance of FedAvg (Karimireddy et al., 2020; Reddi et al., 2020; Li et al., 2020b; Ye et al., 2023b; Li et al., 2021b; Tenison et al., 2022; Wang et al., 2020). Some work has enhanced optimization algorithms by regulating the local training process (Karimireddy et al., 2020; Reddi et al., 2020; Li et al., 2020b; Wang et al., 2020; Li et al., 2021b), while others have proposed alternative aggregation methods beyond simple averaging (Tenison et al., 2022; Ye et al., 2023b). Another line of work focuses on personalized Federated Learning (Smith et al., 2017; Tan et al., 2022; Mansour et al., 2020; Li et al., 2021a; Finn et al., 2017), addressing data heterogeneity by adapting the global model locally for each device.

Aside from data heterogeneity, Wang et al. (2020) notice different local training process (which we refer to training heterogeneity) exacerbates the convergence of federated optimization algorithms, leading them to converge to a stationary point of an objective function inconsistent with the original objective function.

3 Task Arithmetic is One-Shot FedAvg

To deepen our understanding of the mechanism behind Task Arithmetic in multi-task learning, we establish a connection in this section between one-shot FedAvg and Task Arithmetic.

Given T tasks, the objective in multi-task learning is to train a model parameterized by θ that performs well across all T tasks. This can be formulated as minimizing the following multi-task objective function:

$$L(\theta) = \frac{1}{T}\sum_{t=1}^{T} L_t(\theta). \tag{1}$$

Here $L_t(\theta) = \mathbb{E}_{(x_t, y_t) \sim \mathcal{D}_t}[\ell(\theta; x_t, y_t)]$ represents the objective function for task t, where (x_t, y_t) is a pair of input and output drawn from the data distribution $\mathcal{D}_t$, and $\ell(\cdot)$ denotes the loss function associated with the data and model. This formulation aligns with that used in Federated Learning, where each device t has a local objective function L_t. $L(\theta)$ is referred to as the global objective function.

In Federated Learning, the global objective function (1) is often optimized using FedAvg. In FedAvg, each local objective function is optimized through several iterations of Stochastic Gradient Descent (SGD), after which the server averages all the local updates. This process, also known as *local Stochastic Gradient Descent*

(local SGD), is repeated over multiple communication rounds. Formally, given R communication rounds and initial global model parameters θ_0, FedAvg follows the following update process $\forall r \in [R]$:

$$\begin{aligned}
&\theta_{t,r}^{(0)} = \theta_{r-1} \quad \forall t \in [T] \\
&\theta_{t,r}^{(k+1)} = \theta_{t,r}^{(k)} - \eta_{t,r}^{(k)} g_t(\theta_{t,r}^{(k)}) \quad \forall k \in [0, K_t - 1], \forall t \in [T] \\
&\theta_r = \theta_{r-1} + \frac{\beta}{T} \sum_{t=1}^{T} (\theta_{t,r}^{(K_t)} - \theta_{r-1}) = \theta_{r-1} - \frac{\beta}{T} \sum_{t=1}^{T} \sum_{k=0}^{K_t-1} \eta_{t,r}^{(k)} g_t(\theta_{t,r}^{(k)}).
\end{aligned} \tag{2}$$

Here, θ_r represents the parameter of the global objective function at the end of the r-th communication round, while $\theta_{t,r}^{(k)}$ denotes the parameter of the t-th local objective function at the k-th local optimization step during the r-th communication round. The learning rate used for this step is $\eta_{t,r}^{(k)}$. The stochastic gradient of the t-th local objective function L_t is $g_t(\cdot)$, and β is the outer step size used to aggregate all local updates. In the one-shot setting where $R = 1$, the update simplifies to the following:

$$\theta_{OS} = \theta_0 + \frac{\beta}{T} \sum_{t=1}^{T} (\theta_t^{(K_t)} - \theta_0) = \theta_0 - \frac{\beta}{T} \sum_{t=1}^{T} \sum_{k=0}^{K_t-1} \eta_t^{(k)} g_t(\theta_t^{(k)}) \tag{3}$$

where θ_{OS} denotes the parameters generated by one-shot FedAvg.

In Task Arithmetic, the procedure mirrors the process in FedAvg. Each task t independently minimizes its own objective function L_t by performing K_t iterations of SGD with learning rates $\{\eta_t^{(0)}, \ldots, \eta_t^{(K_t-1)}\}$, starting from the same initial model parameters θ_0 and converging to a minimizer $\theta_t^* \in \arg\min L_t(\theta)$. This yields $\theta_t^* = \theta_t^{(0)} - \sum_{k=0}^{K_t-1} \eta_t^{(k)} g_t(\theta_t^{(k)})$ where $\theta_t^{(0)} = \theta_0 \, \forall t$. The task vector τ_t is defined by $\tau_t = \theta_t^* - \theta_t^{(0)}$. Using Task Arithmetic, a new set of parameters can be constructed as

$$\theta_{TA} = \theta_0 + \lambda \sum_{t=1}^{T} \tau_t = \theta_0 - \lambda \sum_{t=1}^{T} \sum_{k=0}^{K_t-1} \eta_t^{(k)} g_t(\theta_t^{(k)}) \tag{4}$$

where λ is a hyperparameter called the scaling coefficient (Ilharco et al., 2023), which controls the extent to which the sum of task vectors is added back to the pre-trained parameters. By comparing equations (3) and (4), we see that performing Task Arithmetic is equivalent to one-shot FedAvg with outer step size $\beta = \lambda T$.

4 Adapting Federated Learning Theory for Task Arithmetic

In this section, we extend theoretical insights from Federated Learning to Task Arithmetic, identifying two main factors that impact its performance: *data heterogeneity* and *training heterogeneity*. We analyze how these factors impact the convergence of Task Arithmetic. In particular, we study its ability to achieve the global minimum of a convex objective function and the local minimum of a non-convex objective function.

4.1 Data Heterogeneity

This subsection is dedicated to understanding how data heterogeneity influences the performance of Task Arithmetic. Data heterogeneity is common in Federated Learning and refers to the situation when data on each device is non-independent and identically distributed (non-i.i.d.) (Li et al., 2020b; Ye et al., 2023a; Wen et al., 2023). In the context of Task Arithmetic, we define data heterogeneity as follows:

Definition 4.1. *(Data Heterogeneity) In Task Arithmetic, data heterogeneity refers to the non-i.i.d nature of training data across different tasks.*

In the convergence analysis of FedAvg, data heterogeneity has been a longstanding issue. Given the connection between FedAvg and Task Arithmetic, in order to understand how data heterogeneity impacts Task Arithmetic, it is helpful to first review existing findings on data heterogeneity in FedAvg. We begin by introducing several standard assumptions commonly used in the literature (Li et al., 2019; Karimireddy et al., 2020; Khaled et al., 2020; Koloskova et al., 2020; Woodworth et al., 2020b;a; Glasgow et al., 2022; Patel et al., 2024; Wang et al., 2022; Woodworth et al., 2021).

Assumption 4.2. *(Convexity and Smoothness) Assume all the task objective functions L_t are convex and H-smooth. That is, $\forall t \in [T]$ and $\forall \theta, \varphi \in \mathbb{R}^d$, $L_t(\theta) \leq L_t(\varphi) + \langle \nabla L_t(\varphi), \theta - \varphi \rangle + \frac{H}{2}\|\theta - \varphi\|^2$.*

Assumption 4.3. *(Bounded Stochastic Noise) The stochastic gradient computed by each task is unbiased with bounded variance. That is, $\forall \theta \in \mathbb{R}^d$,*

$$\mathbb{E}_{(x_t,y_t)\sim\mathcal{D}_t}[\nabla \ell(\theta; x_t, y_t)] = \nabla L_t(\theta) \quad \textit{and} \quad \mathbb{E}_{(x_t,y_t)\sim\mathcal{D}_t}[\|\nabla \ell(\theta; x_t, y_t) - \nabla L_t(\theta)\|^2] \leq \sigma^2.$$

Assumption 4.4. *(Bounded Initialization Error) Assume $\forall \theta^* \in \arg\min_{\theta\in\mathbb{R}^d} L(\theta)$, $\exists B$ such that the initialization θ_0 satisfies $\|\theta_0 - \theta^*\| \leq B$.*

To facilitate the analysis, we assume that all task objective functions are optimized with the same number of iterations, denoted $K_t = K \, \forall t \in [T]$, and that they use constant learning rates $\eta_t^k = \eta \, \forall t \in [T]$ and $\forall k \in [K]$. Additionally, we set the outer step size to $\beta = 1$, reducing Task Arithmetic to model averaging.

Although there is no universal definition of data heterogeneity, several notions are commonly referenced in the literature (Li et al., 2019; Karimireddy et al., 2020; Khaled et al., 2020; Koloskova et al., 2020; Woodworth et al., 2020b;a; Glasgow et al., 2022; Patel et al., 2024; Woodworth et al., 2021). One widely adopted first-order notion of data heterogeneity is given by the following assumption (Koloskova et al., 2020; Patel et al., 2024; Woodworth et al., 2020b; Glasgow et al., 2022).

Assumption 4.5. *(Bounded First-Order Data Heterogeneity at Optima) A set of objective functions $\{L_t\}_{t=1}^T$ satisfies the bounded first-order heterogeneity at optima if $\forall \theta^* \in \arg\min_{\theta\in\mathbb{R}^d} L(\theta)$, $\exists \zeta_*$ such that*

$$\frac{1}{T}\sum_{t=1}^{T} \|\nabla L_t(\theta^*)\|^2 \leq \zeta_*^2.$$

The quantity ζ_*^2 measures the diversity among the set of functions $\{L_t\}_{t=1}^T$ at the optima of the averaged multi-task objective function $L(\theta)$. Here, the notion of data heterogeneity is defined through objective functions, while Ortiz-Jimenez et al. (2023) defines the Task Arithmetic property from the perspective of network functions. In Appendix A, we further explore the connection between this notion of data heterogeneity and the Task Arithmetic property proposed in Ortiz-Jimenez et al. (2023).

Using the notation from Patel et al. (2024), we define a learning problem satisfying Assumptions 4.2, 4.3, 4.4, and 4.5 as belonging to class $\mathcal{P}_{\zeta_*}^{H,B,\sigma}$. Based on the upper bound from Koloskova et al. (2020) and the lower bound from Patel et al. (2024), the following theorem characterizes the convergence of one-shot FedAvg.

Theorem 4.6. *Assume there is only one communication round $R = 1$. Then for any $K \geq 2, T, H, B, \sigma, \zeta_*^2$,*

$$\min_{\{L_t\}_{t=1}^T \in \mathcal{P}_{\zeta_*}^{H,B,\sigma}} \mathbb{E}[L(\theta_{OS})] - L(\theta^*) \succeq HB^2 + \frac{(H\sigma^2 B^4)^{1/3}}{K^{1/3}} + \frac{\sigma B}{\sqrt{TK}} + (H\zeta_*^2 B^4)^{1/3} \tag{5}$$

and

$$\max_{\{L_t\}_{t=1}^T \in \mathcal{P}_{\zeta_*}^{H,B,\sigma}} \mathbb{E}[L(\theta_{OS})] - L(\theta^*) \preceq HB^2 + \frac{(H\sigma^2 B^4)^{1/3}}{K^{1/3}} + \frac{\sigma B}{\sqrt{TK}} + (H\zeta_*^2 B^4)^{1/3}. \tag{6}$$

Here, $\succeq$ and $\preceq$ denote inequalities that hold up to absolute constants. Based on the above theorem, we make several observations about Task Arithmetic. First, data heterogeneity ζ_*^2 degrades the performance of Task Arithmetic. The term $(H\zeta_*^2 B^4)^{1/3}$ is a non-vanishing error term introduced by ζ_*^2, highlighting the negative impact of data heterogeneity.

Second, the one-shot learning nature of Task Arithmetic presents challenges that limit its performance. Notably, another non-vanishing term in Theorem 4.6, HB^2, arises due to the one-shot learning setup. In contrast, for FedAvg with R communication rounds, this term becomes $\frac{HB^2}{R}$ and diminishes as the number of communication rounds R increases. Moreover, although both $\frac{(H\sigma^2 B^4)^{1/3}}{K^{1/3}}$ and $\frac{\sigma B}{\sqrt{TK}}$ decrease as the number of local steps K grows, they decay much more slowly in the one-shot setting compared to R rounds of FedAvg.

With multiple communication rounds, these terms are given by $\frac{(H\sigma^2 B^4)^{1/3}}{K^{1/3}R^{2/3}}$ and $\frac{\sigma B}{\sqrt{TKR}}$ respectively (Patel et al., 2024). This underscores the additional challenges introduced by the one-shot learning paradigm.

Third, starting with a good pre-trained model is important. The influence of pre-training is captured by the term B introduced in Assumption 4.4. This quantity B is a critical factor as it appears in every error term, particularly in the non-vanishing term $(H\zeta_*^2 B^4)^{1/3}$. Starting with a well-suited pre-trained model that has a smaller B significantly mitigates the adverse effects of high data heterogeneity ζ_*^2, as a smaller B counteracts the heterogeneity. In fact, the significance of pre-trained models has been observed in experiments of both Task Arithmetic (Ortiz-Jimenez et al., 2023) and Federated Learning (Nguyen et al., 2022; Chen et al., 2022).

Remark 4.7. **The importance of the scaling coefficient:** As mentioned before, $\lambda = \frac{\beta}{T}$, meaning that the scaling coefficient depends directly on the outer step size. Although in this section we assume $\beta = 1$ which yields $\lambda = \frac{1}{T}$ and reduces Task Arithmetic to model averaging for simplicity, proper tuning of the scaling coefficient λ is essential. Research indicates that the choice of β has a significant impact on FedAvg performance (Patel et al., 2024; Karimireddy et al., 2020; Charles & Konečný, 2020; Jhunjhunwala et al., 2023; Malinovsky et al., 2023; Li et al., 2023b). A similar sensitivity to λ has been observed in Task Arithmetic: the performance of the final model depends heavily on λ. For instance, Figure 15 in Ilharco et al. (2023) illustrates how Task Arithmetic's performance can vary dramatically with changes to λ.

4.2 Training Heterogeneity

This subsection examines the effect of training heterogeneity on Task Arithmetic, which we define as follows:

Definition 4.8. *(Training Heterogeneity) In Task Arithmetic, training heterogeneity refers to the phenomenon where different task objective functions L_t are optimized with varying local learning rates and number of iterations during local training.*

In the previous section, we assumed that all task objective functions are optimized with a fixed number of iterations K and constant learning rate η. However, in practice, each task objective function is often optimized with different hyperparameter settings, which introduces training heterogeneity and can lead to objective inconsistency (Wang et al., 2020). Now, we extend the setting in Section 4.1 to consider each task objective function L_t being optimized with distinct hyperparameters $\eta_t^{(k)}$ and K_t. We adopt notation and apply theoretical insights from Wang et al. (2020) to illustrate the impact of training heterogeneity.

First, let g_t be the stochastic gradient of L_t, and we define the following matrix of stochastic gradients for each task t

$$G_t = [g_t(\theta_t^{(0)}) \quad g_t(\theta_t^{(1)}) \quad \dots \quad g_t(\theta_t^{(K_t-1)})] \in \mathbb{R}^{d\times K_t}.$$

Next we define the vector of normalized learning rates for each task t as

$$a_t = [\frac{\eta_t^{(0)}}{\eta}, \frac{\eta_t^{(1)}}{\eta}, \dots, \frac{\eta_t^{(K_t-1)}}{\eta}]^T \in \mathbb{R}^{K_t}$$

where η is a constant used to normalize the learning rates, whose purpose will be specified later. Using this notation, we can rewrite equation (4) for θ_{TA} as follows:

$$\theta_{TA} = \theta_0 - \lambda \sum_{t=1}^{T} \eta G_t a_t = \theta_0 - \frac{\beta}{T} \sum_{t=1}^{T} \eta \|a_t\|_1 \frac{G_t a_t}{\|a_t\|_1} \tag{7}$$

where the second equality follows from $\lambda = \frac{\beta}{T}$ as mentioned in Section 3. Next, we denote $\tau_{\text{eff}} = \frac{\beta}{T}\sum_{t=1}^{T}\|a_t\|_1$ as the effective number of steps which measures the average number of updates accumulated using the constant learning rate η, and denote $w_t = \frac{\|a_t\|_1}{\sum_{s=1}^{T}\|a_s\|_1}$ as the aggregation weight for task t. Then we can further rewrite equation (7) as

$$\theta_{TA} = \theta_0 - \tau_{\text{eff}} \sum_{t=1}^{T} \eta w_t \frac{G_t a_t}{\|a_t\|_1}. \tag{8}$$

Notice the weight coefficients vector $[w_1; w_2; \ldots; w_T]$ differs from the original uniform coefficients $[\frac{1}{T}; \frac{1}{T}; \ldots; \frac{1}{T}]$ in the objective function L (equation (1)). This discrepancy is caused by training heterogeneity, and leads FedAvg with multiple communication rounds to converge to the stationary point of a different objective function $\tilde{L}(\theta) := \sum_{t=1}^{T} w_t L_t(\theta)$ which is inconsistent with the original objective function L. While Task Arithmetic involves only one round of FedAvg, the inconsistency still remains due to training heterogeneity. Formally, we present the following assumptions and adapt Theorems 1 and 2 from Wang et al. (2020) to illustrate this inconsistency in our setting.

Assumption 4.9. *(Smoothness) Assume all the task objective functions L_t are H-smooth. That is, $\forall t \in [T]$ and $\forall \theta, \varphi \in \mathbb{R}^d$, $\|\nabla L_t(\theta) - \nabla L_t(\varphi)\| \leq H\|\theta - \varphi\|$.*

Assumption 4.10. *(Bounded Gradient Heterogeneity) For any set of weights $\{w_t\}_{t=1}^{T}$ such that $\sum_{t=1}^{T} w_t = 1$, there exist constants α and ζ such that $\forall \theta \in \mathbb{R}^d$,*

$$\sum_{t=1}^{T} w_t \|\nabla L_t(\theta)\|^2 \leq \alpha^2 \|\sum_{t=1}^{T} w_t \nabla L_t(\theta)\|^2 + \zeta^2.$$

Remark 4.11. Notice that Assumption 4.10 imposes a more restrictive condition on data heterogeneity compared to Assumption 4.5. Currently, no unified notion of data heterogeneity exists for Federated Learning. Since this section focuses on training heterogeneity, we adopt this more restrictive notion of data heterogeneity, as done in Wang et al. (2020), to facilitate theoretical development.

Theorem 4.12. *(Theorem 1 and 2 from (Wang et al., 2020)) Consider θ_{TA} from update rule (7). Denote $\tilde{L}(\theta) = \sum_{t=1}^{T} w_t L_t(\theta)$ and $\bar{K} = \frac{1}{T}\sum_{t=1}^{T} K_t$. Let $\eta = \sqrt{T/\bar{K}}$. Under Assumption 4.3, 4.9 and 4.10, we have the following bound on the gradient norm $\|\nabla \tilde{L}(\theta_{TA})\|^2$:*

$$\mathbb{E}[\|\nabla \tilde{L}(\theta_{TA})\|^2] \leq \frac{4(\tilde{L}(\theta_0) - \tilde{L}_{\text{inf}})(\bar{K}/\tau_{\text{eff}})}{\sqrt{T\bar{K}}} + \frac{4H\sigma^2 A_1}{\sqrt{T\bar{K}}} + \frac{6TH^2\sigma^2 A_2}{\bar{K}} + \frac{12TH^2\zeta^2 A_3}{\bar{K}}. \tag{9}$$

Specifically, $\tilde{L}_{\text{inf}} = \inf_\theta \tilde{L}(\theta)$, $A_1 = \tau_{\text{eff}} T \sum_{t=1}^{T} \frac{w_t^2 \|a_t\|_2^2}{\|a_t\|_1^2}$, $A_2 = \sum_{t=1}^{T} w_t(\|a_t\|_2^2 - a_{t,-1}^2)$ and $A_3 = \max_t\{\|a_t\|_1(\|a_t\|_1 - a_{t,-1})\}$ where $a_{t,-1}$ denotes the last coordinate of the vector a_t. Denote the RHS of inequality (9) as ϵ. Moreover, we have the following bound on gradient norm $\|\nabla L(\theta_{TA})\|^2$:

$$\mathbb{E}[\|\nabla L(\theta_{TA})\|^2] \leq 2[\chi^2_{p||w}(\alpha^2 - 1) + 1]\epsilon + 2\chi^2_{p||w}\zeta^2 \tag{10}$$

where $\chi^2_{p||w} = \sum_{t=1}^{T} \frac{(\frac{1}{T} - w_t)^2}{w_t}$ is the chi-square divergence between the weight coefficient vectors $p = [\frac{1}{T}\ \frac{1}{T}\ \ldots\ \frac{1}{T}]$ and $w = [w_1\ w_2\ \ldots\ w_T]$.

The theorem above illustrates how heterogeneous local training affects the convergence of Task Arithmetic to a stationary point of L. The convergence rate in (10) is influenced by two key factors: the term $\chi^2_{p||w}\zeta^2$ and the convergence rate to a stationary point of $\tilde{L}$ in (9). For $\chi^2_{p||w}\zeta^2$, when different training processes are used for each objective function, $\chi^2_{p||w}$ is non-zero, resulting in $\chi^2_{p||w}\zeta^2$ as a persistent error term. With training heterogeneity, $\chi^2_{p||w}\zeta^2$ only vanishes if $\zeta^2 = 0$, indicating minimal data heterogeneity. This highlights the interaction between data heterogeneity and training heterogeneity: significant data heterogeneity exacerbates the negative effects of training heterogeneity, intensifying the overall performance degradation.

When all task objective functions are optimized with the same number of iterations K and a consistent learning rate schedule $\{\eta^{(0)}, \ldots, \eta^{(K-1)}\}$, we have $w_t = \frac{1}{T}$. This yields $\chi^2_{p||w} = 0$ and $L = \tilde{L}$, aligning the actual objective function $\tilde{L}$ being optimized with the original objective function L. In this scenario, objective inconsistency is effectively eliminated. The convergence rate in (10) is reduced to (9).

Remark 4.13. In Theorem 4.12, unlike in Theorem 4.6, we make no assumptions about the convexity of the objective functions, which naturally results in a looser convergence rate. Since the primary focus of this paper is not on deriving a tighter convergence bound for non-convex settings, we limit our analysis to applying existing theoretical results to understand the behavior of Task Arithmetic.

5 Adapting Federated Learning Algorithms for Task Arithmetic

In the previous section, we used insights from FedAvg to analyze how data and training heterogeneity negatively impact Task Arithmetic. In order to address these challenges, numerous algorithms have been developed to improve FedAvg for more efficient Federated Learning (see Section 2 for details). Thus, we adapt some Federated Learning algorithms to solve heterogeneity challenges in Task Arithmetic for better model merging performance. Selecting the right Federated Learning algorithms to implement requires a clear understanding of key challenges that complicate the adaptation. In this section, we begin with analyzing several key challenges.

5.1 Challenges in Adapting Federated Learning Algorithms for Task Arithmetic

First, the number of communication rounds is limited. As Task Arithmetic is only one-shot Federated Learning, algorithms relying on multiple communication rounds are unsuitable. For instance, some Federated Learning algorithms add regularization terms to local objective functions (Li et al., 2020b; Durmus et al., 2021) to encourage local updates to remain close to the global model parameters transmitted from the previous communication round. However, in our one-shot setting, only the pre-trained model parameters θ_0 are communicated, so applying this type of regularization would constrain each task's fine-tuned parameters to be near the pre-trained parameters, potentially degrading both convergence and task-specific performance.

Other algorithms, like those using variance reduction techniques (Karimireddy et al., 2020; Reddi et al., 2020) or adaptively updating the server's optimal outer step size (Jhunjhunwala et al., 2023; Li et al., 2023b), aim to address heterogeneity through an iterative process. These iterative methods require either each local device or the central server to compute, accumulate, and update certain metrics over multiple communication rounds. Since Task Arithmetic operates within a one-shot setting, implementing such iterative updates is impossible. This constraint limits the use of these approaches to modify Task Arithmetic, as they cannot perform the necessary progressive adjustments over time.

Second, no additional training is allowed. To counteract the effects of heterogeneity or limited communication and data budget, some Federated Learning algorithms require additional training, imposing additional computational cost. For instance, Ye et al. (2023b) ask each device to learn metrics that compare local and global data through training, and these metrics are then used as additional scores for aggregation. Zhang et al. (2022) train an additional generator to generate data which is subsequently used to train a global model in one-shot Federated Learning setting.

Third, no additional datasets are available. Many Federated Learning algorithms rely on supplementary datasets, which is not feasible for modifying Task Arithmetic. In one-shot Federated Learning, a common approach to address data heterogeneity is knowledge distillation (Zhou et al., 2020; Guha et al., 2019; Li et al., 2020a). These methods often require access to extra datasets from which either local devices or the central server distills knowledge to improve model performance.

Given the constraints and unique needs for adapting Federated Learning algorithms, we identify the following three criteria for selecting Federated Learning algorithms: adaptability to One-Shot Setting, no additional training required and no access to additional datasets required.

5.2 Algorithms

With criteria established above, we now explore four Federated Learning algorithms FedNova (Wang et al., 2020), FedGMA (Tenison et al., 2022), Median (Yin et al., 2018) and CCLIP (Karimireddy et al., 2021) that align with these guidelines, explaining their motivations and how they modify Task Arithmetic. For more detailed explanation and further understanding of these algorithms, please refer to the original papers.

FedNova (Wang et al., 2020) FedNova addresses objective inconsistency caused by training heterogeneity by replacing the heterogeneous weight vector $[w_1\ w_2\ \dots\ w_T]$ with the uniform weight vector $[\frac{1}{T}\ \frac{1}{T}\ \dots\ \frac{1}{T}]$, ensuring consistent weighting across tasks. This approach adapts easily to a one-shot setting. Using the

notation from Section 4.2, FedNova modifies the Task Arithmetic update as:

$$\theta_{TA} = \theta_0 - \tau_{\text{eff}} \sum_{t=1}^{T} \frac{\eta}{T} \frac{G_t a_t}{\|a_t\|_1} = \theta_0 + \lambda (\frac{1}{T} \sum_{t=1}^{T} \|a_t\|_1) \sum_{t=1}^{T} \frac{\tau_t}{\|a_t\|_1} \tag{11}$$

where $\tau_{\text{eff}} = \frac{\beta}{T} \sum_{t=1}^{T} \|a_t\|_1$ is the effective number of steps defined in Section 4.2, $\tau_t = -\eta G_t a_t$ is the task vector, and $\lambda = \frac{\beta}{T}$ is the scaling coefficient. In other words, FedNova normalizes each task vector by $\|a_t\|_1$ and rescales the scaling coefficient by the average $\frac{1}{T} \sum_{t=1}^{T} \|a_t\|_1$.

FedGMA (Tenison et al., 2022) FedGMA addresses data heterogeneity by mitigating sign conflicts among local updates, which in FedAvg can cause information loss and slower convergence. It uses a gradient mask to reduce the impact of conflicting directions and preserves meaningful information. Specifically, FedGMA computes an agreement score A to measure alignment across task vectors $\{\tau_t\}_{t=1}^{T} \subset \mathbb{R}^d$. Based on a threshold ρ, FedGMA constructs a mask $\tilde{M}$ that emphasizes coordinates with strong agreement while reducing the influence of others. Formally, define $A = \left| \frac{1}{T} \sum_{t=1}^{T} \text{sign}(\tau_t) \right|$ and $\tilde{M}_j = \begin{cases} 1, & \text{if } A_j \geq \rho \\ A_j, & \text{otherwise} \end{cases}$ where j denotes the j-th coordinate and $\text{sign}(\cdot)$ is applied in a coordinate-wise manner. This yields

$$\theta_{TA} = \theta_0 + \lambda \tilde{M} \odot \sum_{t=1}^{T} \tau_t. \tag{12}$$

Median (Yin et al., 2018) Coordinate-Wise Median (Yin et al., 2018), originally designed to handle adversarial updates in Federated Learning, is adapted here to address data and training heterogeneity in Task Arithmetic. Due to diverse data distributions or differing hyperparameter settings, some task vectors may have extreme values. By selecting the median value for each coordinate, this method reduces the influence of outliers while maintaining overall performance across tasks. It modifies Task Arithmetic as

$$\theta_{TA} = \theta_0 + \lambda \, \text{med}(\tau_1, \ldots, \tau_T) \tag{13}$$

where $\text{med}(\cdot)$ computes the coordinate-wise median of $\{\tau_t\}_{t=1}^{T}$.

CCLIP (Karimireddy et al., 2021) CCLIP, short for centered clipping, is another widely applied robust aggregation method towards adversarial devices in Federated Learning. With the same motivation to using the Coordinate-Wise Median, we use CCLIP to reduce the impact of extreme task vectors. CCLIP is implemented with a predefined threshold ρ and modifies Task Arithmetic as follows:

$$\theta_{TA} = \theta_0 + \lambda \sum_{t=1}^{T} \tau_t \min\{1, \frac{\rho}{\|\tau_t\|}\}. \tag{14}$$

When the norm of a task vector $\|\tau_t\|$ exceeds the threshold ρ, this method identifies it as an outlier and shrinks its magnitude by a factor of $\frac{\rho}{\|\tau_t\|}$.

6 Experiments on Merging CLIP

In this section, we present and discuss our experimental results on CLIP-ViT-B-32 (Radford et al., 2021) for image classifications. We follow the same experimental paradigm as (Ilharco et al., 2023). Specifically, we use CLIP-ViT-B-32 (Radford et al., 2021) as the pre-trained model and eight datasets: Cars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Stallkamp et al., 2011), MNIST (LeCun, 1998), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2016), and SVHN (Netzer et al., 2011), to construct eight task vectors.

In total, there are 247 ways to select T different task vectors from these eight task vectors where $T \in [2, 8]$: $247 = \sum_{T=2}^{8} \binom{8}{T}$. For each algorithm, we therefore conduct 247 experiments. In each experiment, we merge T selected task vectors and evaluate on the T datasets corresponding to the task vectors used. Our evaluation

metric is normalized accuracy (Ilharco et al., 2023), defined as the test accuracy normalized by the fine-tuned model's accuracy. That is,

$$\text{normalized accuracy on task } t = \frac{\text{accuracy on task } t}{\text{accuracy of the fine-tuned model } t \text{ on task } t}.$$

6.1 Experimental Results

In the first part of the experiments (Section 6.1.1), to simulate practical conditions of training heterogeneity, we fine-tune CLIP-ViT-B-32 on each dataset using three learning rates $\{1e{-}4, 1e{-}5, 1e{-}6\}$ and different numbers of iterations. Then we select the best fine-tuned checkpoints via cross-validation on validation datasets. Refer to Appendix B.1 for further details on fine-tuning and cross-validation.

In the second part of the experiments (Section 6.1.2), to better understand the impact of training heterogeneity, we use the task vectors provided by Ilharco et al. (2023), which were fine-tuned with uniform training conditions—the same number of iterations and the same learning rates—thereby eliminating training heterogeneity.

6.1.1 Merging with training heterogeneity

We first report experimental results using task vectors fine-tuned with training heterogeneity. Table 1 summarizes the performance of various methods in a specific experimental setup: merging all eight task vectors, corresponding to the scenario where $T = 8$. We report the average normalized accuracy as well as the normalized accuracy for each dataset. Task Arithmetic is used as the baseline method for comparison. As shown in the table, all four adapted Federated Learning methods outperform the baseline by a substantial margin. Moreover, we observe that Median and CCLIP yield the most improvement.

Methods	Average Normalized Accuracy	DTD	EuroSAT	GTSRB	SUN397	SVHN	MNIST	Cars	RESISC45
Task Arithmetic	67.33	57.43	53.86	41.00	82.40	**78.58**	87.76	71.94	65.72
Median	<u>74.55</u> (↑ 7.22)	**67.51**	**78.05**	<u>67.12</u>	84.02	56.69	<u>91.32</u>	**77.51**	**74.17**
FedNova	69.57 (↑ 2.24)	57.08	50.37	61.47	**86.62**	<u>77.68</u>	85.18	74.22	63.96
FedGMA	68.55 (↑ 1.22)	60.01	58.69	45.02	<u>84.13</u>	71.19	86.65	74.53	68.13
CCLIP	**74.82** (↑ 7.49)	<u>66.76</u>	<u>75.42</u>	**73.87**	83.18	58.51	**92.03**	<u>76.40</u>	<u>72.39</u>

Table 1: **Combining all eight task vectors using five different methods.** Each method is evaluated on eight datasets, with normalized accuracy reported for each dataset. The highest and second-highest normalized accuracy values for each dataset are highlighted in bold and underlined, respectively.

Methods	Percentage of Improved Experiments Compared to Task Arithmetic	Percentage of Unchanged Experiments Compared to Task Arithmetic	Percentage of Degraded Experiments Compared to Task Arithmetic
Median	67.61%	0%	32.39%
FedNova	63.56%	0%	36.44%
FedGMA	40.49%	18.22%	41.29%
CCLIP	91.50%	0%	8.5%

Table 2: **Percentage of improved, unchanged, and degraded experiments using different methods compared to Task Arithmetic.**

Table 2 and Figure 1 summarize the performance comparison between Task Arithmetic and other Federated Learning algorithms across 247 experiments. In Table 2, we report the percentage of 247 experiments in which the average normalized accuracy improves, remains unchanged, or degrades when using four Federated Learning methods compared to the baseline, Task Arithmetic. The average normalized accuracy is calculated by averaging over the number of task vectors being used. In order to better visualize the performance differences for each method, in Figure 1, we use histograms to show the frequencies of experiments within each range of change in average normalized accuracy. Median, FedNova, and CCLIP consistently show the

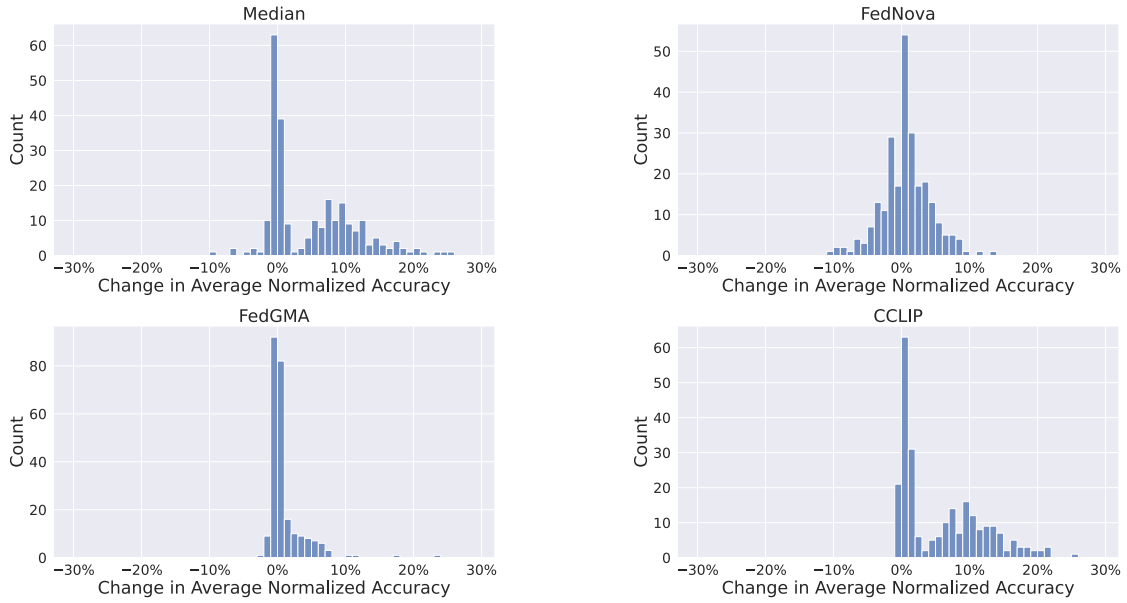

Figure 1: **Histograms showing the change in average normalized accuracy for four different methods compared to Task Arithmetic.** The x-axis shows the difference in average normalized accuracy relative to Task Arithmetic, with positive values indicating improvement and negative values indicating degradation. The y-axis represents the number of experiments within each range of change values.

ability to improve upon Task Arithmetic in most cases, while FedGMA typically demonstrates either no change or slight improvements.

6.1.2 Merging without training heterogeneity

In Section 4.2, we analyzed how training heterogeneity causes objective inconsistency, degrading Task Arithmetic's performance. While in the experiments conducted by Ilharco et al. (2023), all task vectors were homogeneously fine-tuned using a consistent learning rate 1e−5, our approach in Section 6.1.1 employs heterogeneous fine-tuning. We compare the performance of Task Arithmetic on these two sets of task vectors to validate our theoretical findings in Section 4.2.

Table 3 compares Task Arithmetic's performance on heterogeneously and homogeneously fine-tuned task vectors when merging all eight task vectors. Again we report the average normalized accuracy and the normalized accuracy for each dataset. As evident from the table, Task Arithmetic with homogeneous fine-tuning consistently outperforms its heterogeneous counterpart across all datasets, except for SUN397.

Table 4 and Figure 2 compare Task Arithmetic's performance on homogeneously and heterogeneously fine-tuned task vectors across 247 experiments. Homogeneous fine-tuning outperforms heterogeneous fine-tuning in 92.31% of cases, as shown in Table 4. Moreover, Figure 2 shows that homogeneous fine-tuning can improve the average normalized accuracy by up to more than 30%. These results highlight the significant negative impact of training heterogeneity on the performance of Task Arithmetic.

6.2 Discussion on Experimental Results

We now discuss a key observation from our experimental results: in practice, training heterogeneity poses a greater challenge than data heterogeneity for Task Arithmetic.

While Section 4.1 highlights how data heterogeneity degrades Task Arithmetic, our experimental results in Section 6.1 show it is less problematic compared to training heterogeneity. First, FedNova, designed to address training heterogeneity, consistently outperforms Task Arithmetic more frequently and significantly

Methods	Average Normalized Accuracy	DTD	EuroSAT	GTSRB	SUN397	SVHN	MNIST	Cars	RESISC45
Task Arithmetic with Heterogeneous Fine-Tuning	67.33	57.43	53.86	41.00	82.40	78.58	87.76	71.94	65.72
Task Arithmetic with Homogeneous Fine-Tuning	77.34 (↑ 10.01)	64.90	77.93	69.47	80.64	80.26	96.42	75.98	73.01

Table 3: **Using Task Arithmetic to combine eight task vectors from heterogeneous and homogeneous fine-tuning processes.** Each method is evaluated on eight datasets, with normalized accuracy reported for each dataset.

	Percentage of Improved Experiments	Percentage of Unchanged Experiments	Percentage of Degraded Experiments
Task Arithmetic with Homogeneous Fine-Tuning	92.31%	0%	7.69%

Table 4: **Percentage of improved, unchanged, and degraded experiments using task vectors with homogeneous fine-tuning process compared to those with heterogeneous fine-tuning process.**

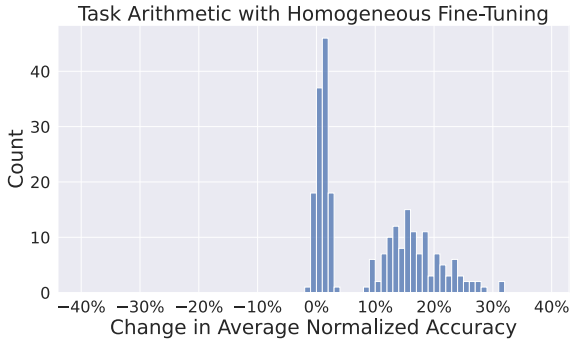

Figure 2: **Histogram showing the change in average normalized accuracy when using task vectors from homogeneous fine-tuning compared to heterogeneous fine-tuning.** The x-axis shows the difference in average normalized accuracy relative to Task Arithmetic, with positive values indicating improvement and negative values indicating degradation. The y-axis represents the number of experiments within each range of change values.

than FedGMA, which targets data heterogeneity. Table 2 and Figure 1 demonstrate that FedNova both improves performance more frequently and achieves greater overall gains than FedGMA.

Second, among Federated Learning algorithms, CCLIP and Median demonstrate the best performance. As discussed in Section 5, these methods are designed for robust aggregation in the presence of outliers. In our setting, they effectively address training heterogeneity which causes certain task vectors to have disproportionately large norms and behave like outliers. For example, the cross-validation process selects a much larger learning rate of 1e−4 for SVHN, compared to 1e−5 used for other datasets. This hyperparameter setup results in the SVHN task vector having a significantly larger norm (reported in Appendix B.1.5), making it an outlier that negatively impacts the merged model's performance on other tasks when using Task Arithmetic. By employing robust aggregation methods like Median and CCLIP, we reduce the influence of the SVHN task vector, which improves the merged model's performance on other tasks.

Third, when comparing Table 2 and Table 4, we see that homogeneous fine-tuning leads to more frequent improvements over Task Arithmetic compared to the other four algorithms Median, FedNova, FedGMA and CCLIP. Similarly, Figure 2 demonstrates that homogeneous fine-tuning results in the most frequent and substantial positive changes in average normalized accuracy.

Further evidence is presented in Appendix B.2, where we evaluate the performance of Median, FedGMA and CCLIP on task vectors generated via homogeneous fine-tuning. Using the performance of Task Arithmetic on these homogeneously fine-tuned task vectors as the baseline, we find that Federated Learning algorithms rarely improve upon the baseline. In fact, Task Arithmetic consistently emerges as the best-performing approach when merging these task vectors generated without training heterogeneity. This reinforces our observation that training heterogeneity is a more significant issue than data heterogeneity in practice.

7 Experiments on Merging LLMs

We now present and discuss our experimental results on merging LLMs for three tasks: instruction following, mathematical reasoning, and code generation. We follow the experimental paradigm of Yu et al. (2024). We merge task vectors from three models—WizardLM-13B (Xu et al., 2023), WizardMath-13B (Luo et al., 2023), and Llama-2-13B-Code-Alpaca (Chaudhary, 2023)—for instruction following, mathematical reasoning, and code generation, respectively. All three models are fine-tuned from Llama2-13B (Touvron et al., 2023). For instruction following, we evaluate the models on AlpacaEval (Li et al., 2023a). For mathematical reasoning, we use GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). For code generation, we use HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). Performance metrics include win rate for AlpacaEval, zero-shot accuracy for GSM8K and MATH, and pass@1 for HumanEval and MBPP.

Since all models used in this experiment are downloaded from HuggingFace, we do not have access to their fine-tuning hyperparameter settings. As a result, FedNova cannot be applied in this experiment because it requires knowledge of learning rates and the number of iterations, which are unavailable. Furthermore, when implementing Median, taking the median of two vectors reduces to averaging, which is equivalent to Task Arithmetic. Consequently, we implement Median only for merging three task vectors, with the corresponding results deferred to Appendix C.2. For additional details on experiments, please refer to Appendix C.

7.1 Experimental Results

In Table 5, we compare the performance of three methods: Task Arithmetic, FedGMA, and CCLIP. The results show that when merging two out of three task vectors, FedGMA and CCLIP often outperform Task Arithmetic. However, when merging all three task vectors, Task Arithmetic shows superior performance on code generation and instruction-following tasks. Notably, Task Arithmetic consistently excels in instruction-following tasks, achieving either the highest accuracy or accuracy comparable to the other methods.

Tasks	Methods	Mathematical Reasoning		Code Generation		Instruction Following
		GSM8K	MATH	HumanEval	MBPP	AlpacaEval
Math Code	Task Arithmetic	64.22	**14.1**	1.22	8.66	/
	FedGMA	65.5	12.66	**15.85**	**21.8**	/
	CCLIP	**65.81**	13.48	4.27	7.6	/
Instruction Math	Task Arithmetic	65.88	13.32	/	/	69.96
	FedGMA	**66.72**	**14.48**	/	/	62.04
	CCLIP	64.75	13.18	/	/	**69.99**
Instruction Code	Task Arithmetic	/	/	**32.32**	32.2	**79.76**
	FedGMA	/	/	20.12	26	49.55
	CCLIP	/	/	**32.32**	**34.2**	76.02
Instruction Math Code	Task Arithmetic	58.45	12.06	**25.16**	**31**	**70.89**
	FedGMA	57.16	11.96	20.12	27.4	64.13
	CCLIP	**62.93**	**12.96**	20.12	27.6	66.91

Table 5: **Performance of merging LLMs.** The best performance for each dataset is highlighted in bold.

7.2 Discussion on Experimental Results

In this section, we discuss a key observation from our experimental results: training heterogeneity arises not only from differences in hyperparameters but also from variations in tuning methods.

In Section 4.2, we theoretically analyzed how using different learning rates and number of iterations creates training heterogeneity and thus leads to objective inconsistency. However, our experimental results in Section 7 reveal that employing different fine-tuning methods further exacerbates training heterogeneity. For instance, in our experiments, the Llama-2-13B-Code-Alpaca model is fine-tuned using QLoRA (Dettmers et al., 2024), a parameter-efficient fine-tuning (PEFT) approach.

PEFT adjusts only a small subset of parameters while leaving the rest unchanged (Han et al., 2024). Consequently, task vectors generated by PEFT typically have smaller norms compared to those generated through standard fine-tuning. This discrepancy can pose challenges when merging task vectors. Simply regulating the behavior of task vectors with larger norms can lead to unintended negative effects, and, to date, no Federated Learning or model merging algorithm has been specifically designed to address this issue.

In our experiments, Llama-2-13B-Code-Alpaca, which is fine-tuned for code generation using PEFT, produces a task vector with a notably small norm of 5.05. In contrast, WizardLM-13B and WizardMath-13B, fine-tuned for instruction following and mathematical reasoning via standard fine-tuning, generate task vectors with much larger norms of 142.61 and 52.62, respectively. This significant disparity in task vector norms between code generation and instruction following leads to complications when merging task vectors by using Federated Learning algorithms. As shown in Table 5, when merging tasks include both instruction following (large norm) and code generation (small norm), FedGMA and CCLIP either fail to outperform Task Arithmetic or achieve comparable performance on these two tasks. This highlights that addressing training heterogeneity by focusing solely on differences in hyperparameters is insufficient in practice.

While some studies have explored the challenges of merging large models fine-tuned via PEFT (Zhang et al., 2023; Wu et al., 2024), merging PEFT-generated task vectors with those produced by standard fine-tuning remains an open research question. Further investigation is required to devise effective strategies for combining such task vectors in Task Arithmetic. Additionally, more research is needed to develop robust aggregation methods in Federated Learning to address this type of practical training heterogeneity.

8 Conclusion

In this paper, we establish a connection between Task Arithmetic and one-shot Federated Learning. By leveraging theoretical insights from Federated Learning, we identify and analyze two key sources of heterogeneity—data heterogeneity and training heterogeneity—and their impact on Task Arithmetic. Also, we adapt Federated Learning algorithms for model merging, demonstrating their great potential to significantly improve over the performance of Task Arithmetic. We hope this work serves as a foundation for advancing the understanding, enhancing the algorithms, and expanding the applications of Task Arithmetic through the lens of Federated Learning.

References

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Gregory Benton, Wesley Maddox, Sanae Lotfi, and Andrew Gordon Gordon Wilson. Loss surface simplexes for mode connecting volumes and fast ensembling. In *International Conference on Machine Learning*, pp. 769–779. PMLR, 2021.

Zachary Charles and Jakub Konečnỳ. On the outsized importance of learning rates in local update methods. *arXiv preprint arXiv:2007.00878*, 2020.

Sahil Chaudhary. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca, 2023.

Hong-You Chen, Cheng-Hao Tu, Ziwei Li, Han-Wei Shen, and Wei-Lun Chao. On the importance and applicability of pre-training for federated learning. *arXiv preprint arXiv:2206.11488*, 2022.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. Cold fusion: Collaborative descent for distributed multitask finetuning. *arXiv preprint arXiv:2212.01378*, 2022.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.

Alp Emre Durmus, Zhao Yue, Matas Ramon, Mattina Matthew, Whatmough Paul, and Saligrama Venkatesh. Federated learning based on dynamic regularization. In *International conference on learning representations*, 2021.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020a.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3259–3269. PMLR, 13–18 Jul 2020b. URL https://proceedings.mlr.press/v119/frankle20a.html.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

Margalit R Glasgow, Honglin Yuan, and Tengyu Ma. Sharp bounds for federated averaging (local sgd) and continuous perspective. In *International Conference on Artificial Intelligence and Statistics*, pp. 9050–9090. PMLR, 2022.

Xinran Gu, Kaifeng Lyu, Longbo Huang, and Sanjeev Arora. Why (and when) does local sgd generalize better than sgd? *arXiv preprint arXiv:2303.01215*, 2023.

Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

Vipul Gupta, Santiago Akle Serrano, and Dennis DeCoste. Stochastic weight averaging in parallel: Large-batch training that generalizes well. *arXiv preprint arXiv:2001.02312*, 2020.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Fedexp: Speeding up federated averaging via extrapolation. *arXiv preprint arXiv:2301.09604*, 2023.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning*, pp. 5311–5319. PMLR, 2021.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pp. 4519–4529. PMLR, 2020.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pp. 5381–5393. PMLR, 2020.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Qinbin Li, Bingsheng He, and Dawn Song. Practical one-shot federated learning for cross-silo setting. *arXiv preprint arXiv:2010.01017*, 2020a.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10713–10722, 2021a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020b.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021b.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023a.

Zexi Li, Tao Lin, Xinyi Shang, and Chao Wu. Revisiting weighted aggregation in federated learning with neural networks. In *International Conference on Machine Learning*, pp. 19767–19788. PMLR, 2023b.

Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

Grigory Malinovsky, Konstantin Mishchenko, and Peter Richtárik. Server-side stepsizes and sampling without replacement provably help in federated optimization. In *Proceedings of the 4th International Workshop on Distributed Machine Learning*, pp. 85–104, 2023.

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17703–17716. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/70c26937fbf3d4600b69a129031b66ec-Paper-Conference.pdf.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

John Nguyen, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. Where to begin? on the impact of pre-training and initialization in federated learning. *arXiv preprint arXiv:2206.15387*, 2022.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 66727–66754. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d28077e5ff52034cd35b4aa15320caea-Paper-Conference.pdf.

Kumar Kshitij Patel, Margalit Glasgow, Ali Zindari, Lingxiao Wang, Sebastian U Stich, Ziheng Cheng, Nirmit Joshi, and Nathan Srebro. The limits and potentials of local sgd for distributed heterogeneous learning with intermittent communication. *arXiv preprint arXiv:2405.11667*, 2024.

Angelo Porrello, Lorenzo Bonicelli, Pietro Buzzega, Monica Millunzi, Simone Calderara, and Rita Cucchiara. A second-order perspective on compositionality and incremental learning. *arXiv preprint arXiv:2405.16350*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In *International Conference on Machine Learning*, pp. 9722–9732. PMLR, 2021.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pp. 1453–1460. IEEE, 2011.

George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. Zipit! merging models from different tasks without training. *arXiv preprint arXiv:2305.03053*, 2023.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12):9587–9603, 2022.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Irene Tenison, Sai Aravind Sreeramadas, Vaikkunth Mugunthan, Edouard Oyallon, Irina Rish, and Eugene Belilovsky. Gradient masked averaging for federated learning. *arXiv preprint arXiv:2201.11986*, 2022.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33: 7611–7623, 2020.

Jianyu Wang, Rudrajit Das, Gauri Joshi, Satyen Kale, Zheng Xu, and Tong Zhang. On the unreasonable effectiveness of federated averaging with heterogeneous data. *arXiv preprint arXiv:2206.04723*, 2022.

Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, and Wensheng Zhang. A survey on federated learning: challenges and applications. *International Journal of Machine Learning and Cybernetics*, 14(2): 513–535, 2023.

Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local sgd better than minibatch sgd? In *International Conference on Machine Learning*, pp. 10334–10343. PMLR, 2020a.

Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33:6281–6292, 2020b.

Blake E Woodworth, Brian Bullins, Ohad Shamir, and Nathan Srebro. The min-max complexity of distributed stochastic convex optimization with intermittent communication. In *Conference on Learning Theory*, pp. 4386–4437. PMLR, 2021.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022a.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7959–7971, 2022b.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*, 2024.

Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119:3–22, 2016.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36, 2024.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. *arXiv preprint arXiv:2310.02575*, 2023.

Guandao Yang, Tianyi Zhang, Polina Kirichenko, Junwen Bai, Andrew Gordon Wilson, and Chris De Sa. Swalp: Stochastic weight averaging in low precision training. In *International Conference on Machine Learning*, pp. 7015–7024. PMLR, 2019.

Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023a.

Rui Ye, Mingkai Xu, Jianyu Wang, Chenxin Xu, Siheng Chen, and Yanfeng Wang. Feddisco: Federated learning with discrepancy-aware collaboration. In *International Conference on Machine Learning*, pp. 39879–39902. PMLR, 2023b.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pp. 5650–5659. Pmlr, 2018.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. Dense: Data-free one-shot federated learning. *Advances in Neural Information Processing Systems*, 35: 21414–21428, 2022.

Jinghan Zhang, Junteng Liu, Junxian He, et al. Composing parameter-efficient modules with arithmetic operation. *Advances in Neural Information Processing Systems*, 36:12589–12610, 2023.

Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020.

Tongtian Zhu, Fengxiang He, Kaixuan Chen, Mingli Song, and Dacheng Tao. Decentralized sgd and average-direction sam are asymptotically equivalent. In *International Conference on Machine Learning*, pp. 43005–43036. PMLR, 2023.